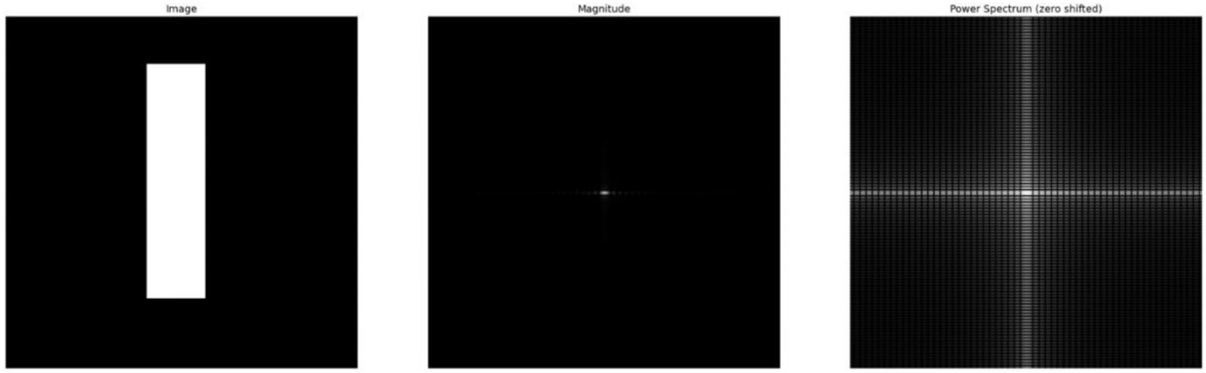
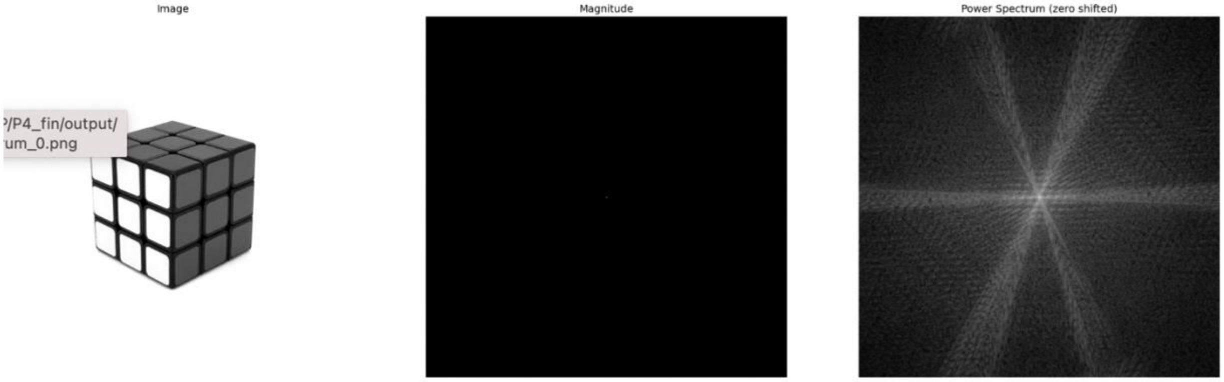
Image Processing Project 4

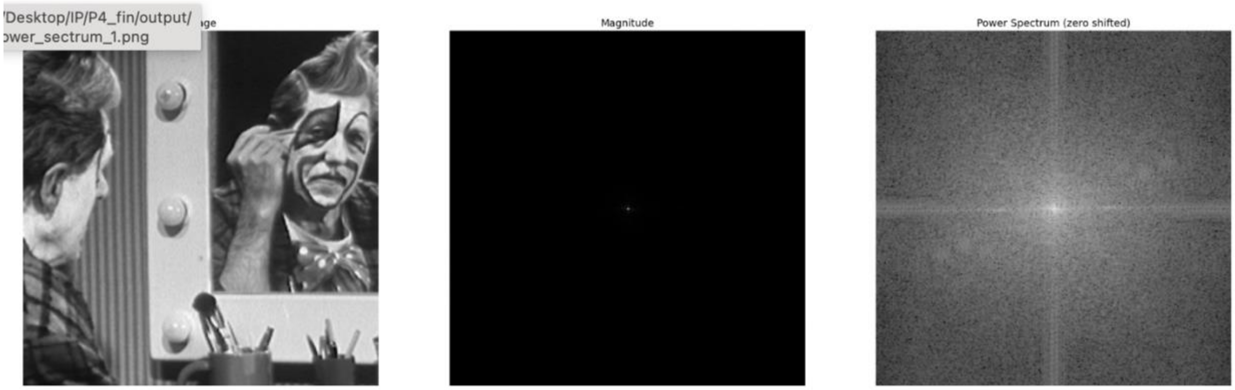
Project report by Yash Lad – u1414500

Part 1:
Fourier Transform and Power Spectrum

In the Powe spectrum (log(magnitude plot)), we can see that for the 1st image it has two strong central lines. These correspond to the edges of the rectangle. We can infer that we need strong frequencies along the directions of the edges to reconstruct the image. Similarly for the 2nd image we can see that it has 3 strong lines, meaning high frequencies along those 3 lines. In the cube image we can see that it has dominant edges along three directions, which explain the magnitude plot. For a more complex image like the 3rd one, it's difficult to fully comprehend the magnitude plot of the Fourier Transform.

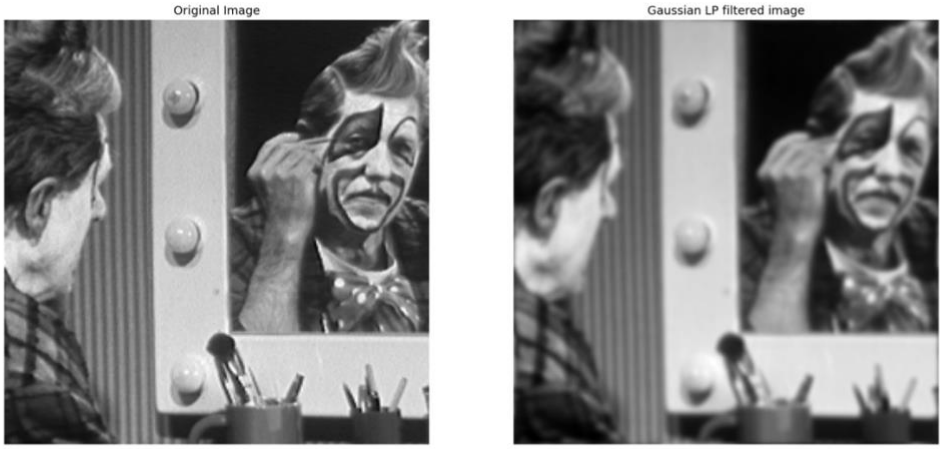


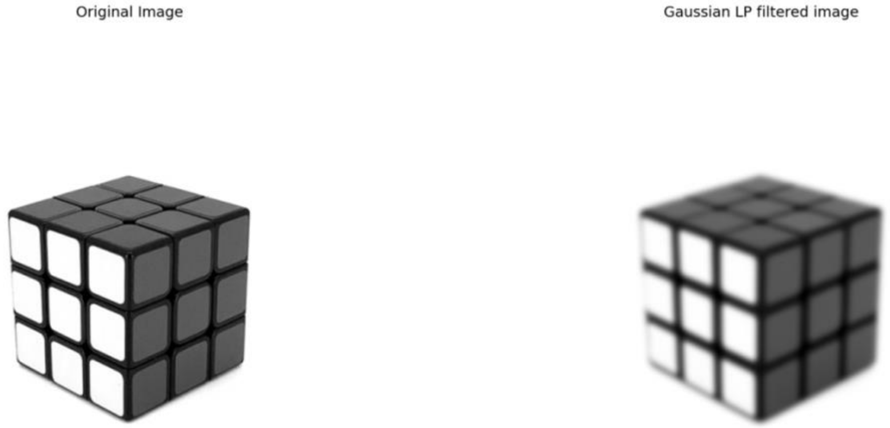




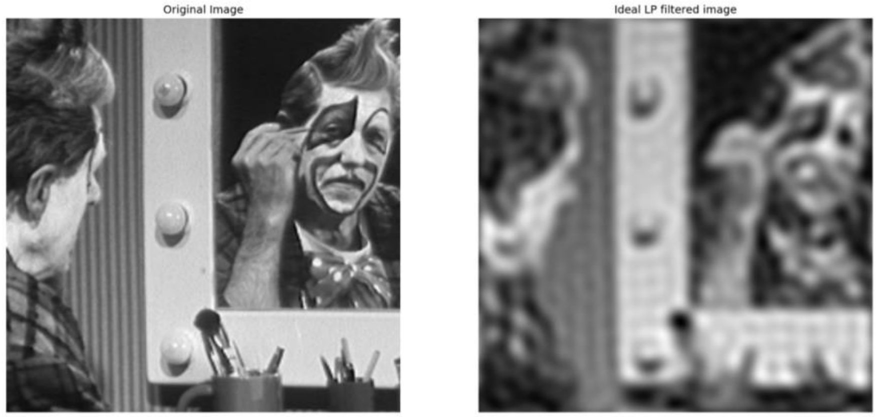
Low Pass filtering:

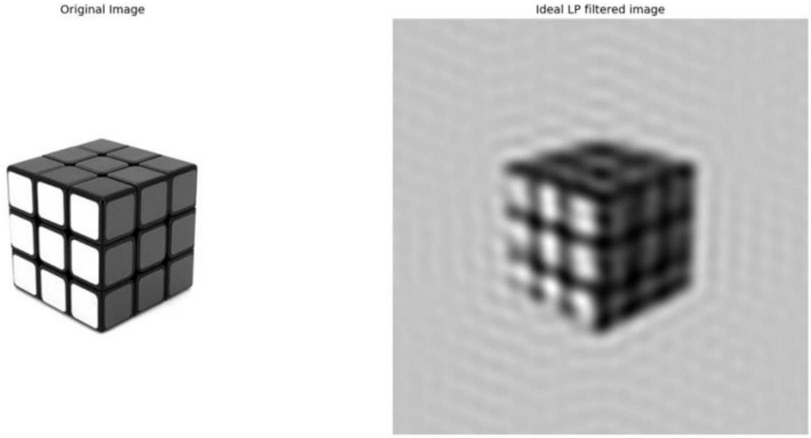
1. Gaussian Low pass filter:





2. Ideal Low pass filter:



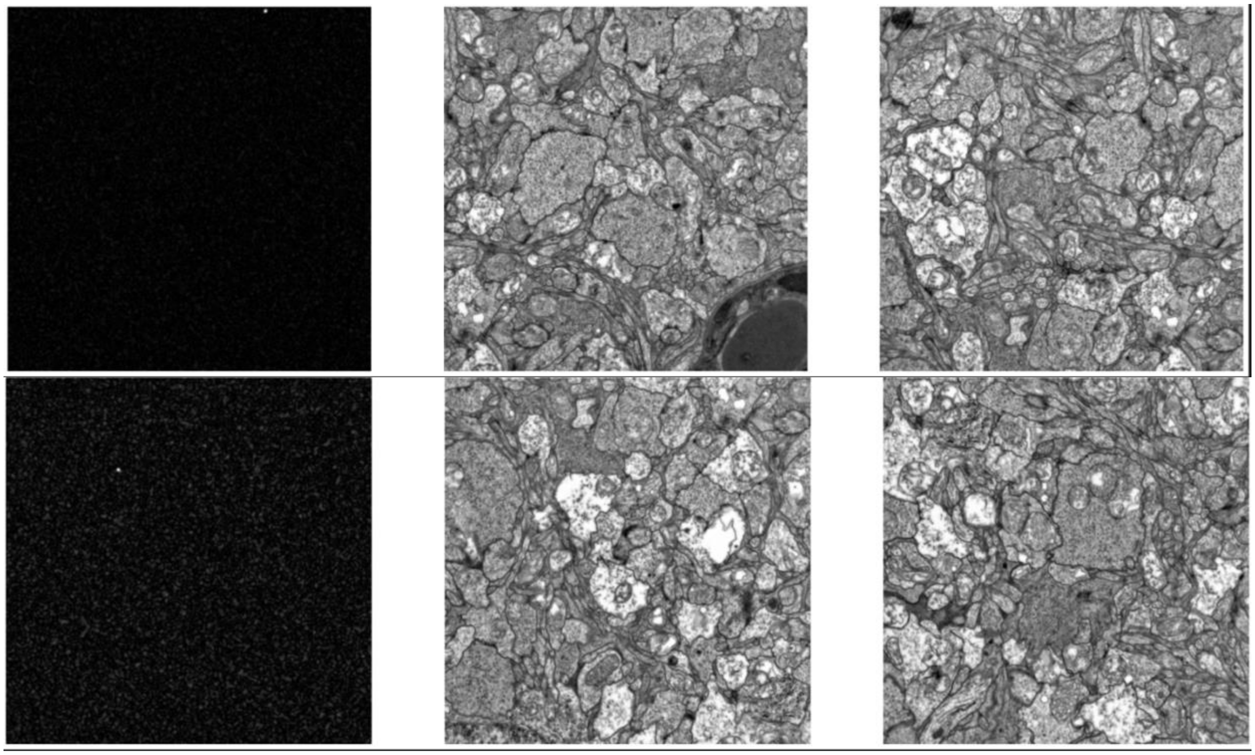


We can see that gaussian low pass filter performs much better than the ideal low pass filter, which blocky leaves artifacts on the filtered image.

Part 2: Image Mosaicking

1. Phase Correlation:

Built a routine that calculates the phase correlation between two images, and returns its peak location. We can see a single maximum peak value in the images below. For my implementation, I took the peak value where the intensity is maximum in the phase correlated image.



2. Image shift implementation:

For any given peak, for a pair of images, these are 4 possible locations of how the two images might be placed with respect to each other:

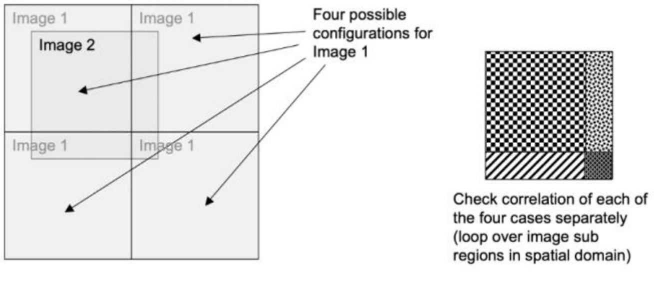


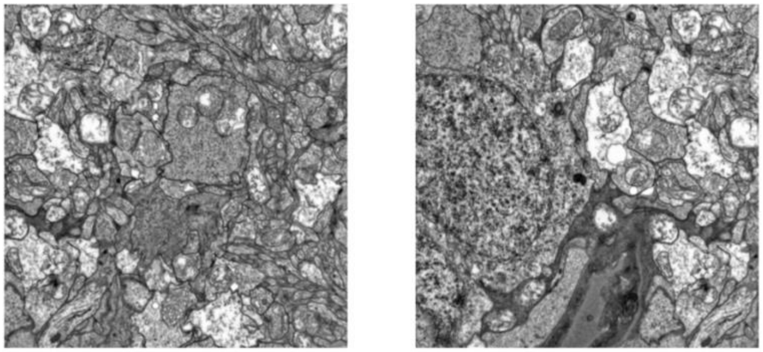
Image 1
Image 1
Image 2
Image 1
Image 1
Four possible configurations for Image 1
Check correlation of each of the four cases separately (loop over image sub regions in spatial domain)

For my implementation, I have placed the reference image, duplicated on edges along three side like in the image above.

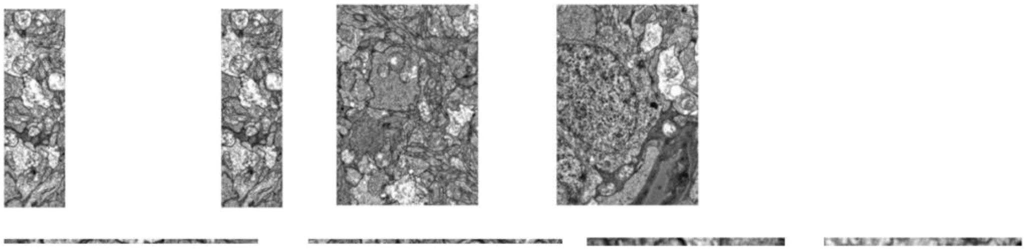
Next I look at the regions corresponding to the 4 overlapping regions between the two images when the second image is placed at the peak location.

For each overlapping region, I compute the phase correlation between the overlapping parts and take the region which gives the maximum phase correlation. Then I compute the shift coordinate that must be a applied to the image when placing it on canvas over the reference image based on the region which gives the maximum value.

The two images:



4 Overlapping regions:

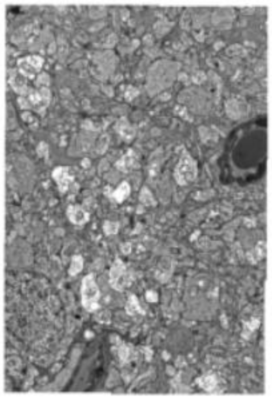


3. Mosaic Building:

Algorithm :

1. Create an empty canvas with white background
2. Maintain a list of images already placed on the canvas
3. Get the image which is correlated with most other images above a certain threshold place this image on the canvas as a starting point
4. Find the image which is best correlated with this image, calculate the shift coordinates, and place this image on the canvas. Add it to placed list.
5. Take the image previously placed on the canvas as a reference and repeat step 4.
6. Loop until all images are placed on the canvas.

Mosaic Cell Image with my implementation:



References:
https://numpy.org/doc/stable/reference/generated/numpy.absolute.html
https://numpy.org/doc/stable/reference/generated/numpy.fft.ifftshift.html